Diagnosing Common Application Problems Under LE on VM/ESA

Session 9124

Mark Picard
LE Level 2 Support
picard@us.ibm.com

July 2000

Agenda

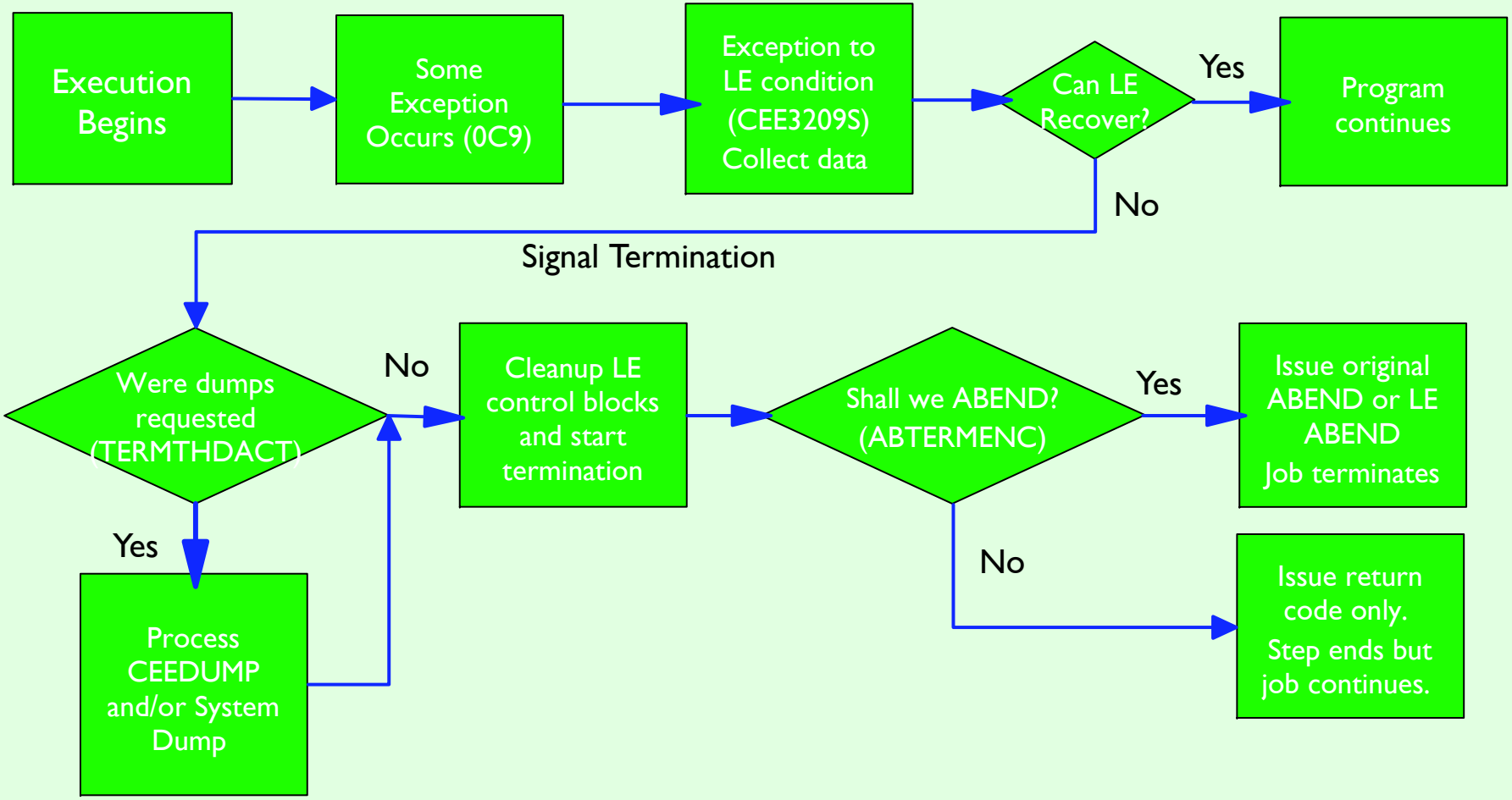
- Common LE functions
- LE Condition Handling
- Important Modules
- Messages and ABENDs
- Condition Tokens
- Storage problems?
- Collecting Error Documentation
- Diagnosing Application Problems
- Summary
- Additional Information

Common LE Functions

- CEL is a set of common functions and routines used by all member languages of LE
  - Initialization
  - Storage management
  - ***Condition Handling***
  - Message services
  - Date/Time services
  - Math functions
  - Termination

LE Condition Handling

- Condition Handling Flow (extremely over simplified)



Important Modules

- CEEHDSP
  - Always the top CSECT in CEEDUMPs
  - Schedules the CEEDUMP to be taken and schedules termination - **Ignore**
    - LE Condition Handling CSECTs start with CEEH*
- CEEPLPKA
  - LE's main load module, contains CEEHDSP
  - ABENDs reported here can be either LE or application failures

Important Modules

- CEEBINIT
  - LE's main initialization load module
    - LE Initialization CSECTs start with CEEBxxxx
- CEEHSGLT
  - Signal processor with termination
    - LE's way of raising a condition
    - **IGNORE** - the previous module detected the problem

Important Modules

- CEEEVxxx
  - LE Event Handler load modules
  - xxx is the member number of the language
    - 003 C/C++
    - 005 COBOL
    - 007 FORTRAN
    - 008 DCE
    - 010 PL/I
    - 011 VA PL/I
    - 012 Debug Tool

Messages and ABENDs

- Message (and Module) Prefixes
  - CEE CEL (but may be reporting a problem elsewhere)
  - IGZ COBOL
  - IBM PL/I
  - AFH FORTRAN
  - EDC C/C++
- See LE Debug Guide and Run-Time Messages for details on LE messages and ABENDs

Messages and ABENDs

- Common CEL messages
  - CEE32nnS Indicates ABEND0Cx
  - CEE3250C Some non-0Cx ABEND occurred
  - CEE0802C HEAP damage detected (normally application error)
  - CEE0813S Out of storage
  - CEE0374C LE detected an error (examine token)

Messages and ABENDs

- U4038 Some Severe error occurred, but no dump was requested (useless)
- U4039 Some Severe error occurred and a CEEDUMP (and optionally System dump) was requested
- U4083 Savearea backchain in error*
- U4087 Error during error processing*
- U4093 Error during initialization*
- U4094 Error during termination*

* reason code needed to be meaningful

Condition Tokens

- Condition Token (Feedback Code)
  - Example: 00030C84 59C3C5C5 xxxxxxxx
    - 0003 | 0C84 | 59 | C3C5C5 | xxxxxxxx
      - 0003 Severity
        - 0000 Informational (I)
        - 0001 Warning (W)
        - 0002 Error (E)
        - 0003 Severe (S)
        - 0004 Critical (C)

Condition Tokens

- Condition Token *(continued)*
  - 0003 | 0C84 | 59 | C3C5C5 | xxxxxxxx
    - 0C84 Hex message number (3204)
    - 59 Flags (ignore)
    - C3C5C5 Facility ID (message prefix)
    - xxxxxxxx Instance specific information (internal use)
  - This token represents message CEE3204S (0C4)

Storage Problems?

- CEE0813S or other out of storage indicators
  - Use RPTSTG RTO to adjust HEAP and STACK allocations
  - Move application above the line when possible - set ALL31(ON), etc
- poor performance due to storage many obtains
  - use HEAPPOOLS RTO - predominately helpful for C/C++ malloc
  - adjust HEAP/STACK increments

Storage Problems? (cont)

- HEAP damage - CEE0802S or other indicators
  - almost always an application problem - may not have manifest on other 'platforms'
  - System dump critical for debug
  - Use HEAPCHK(ON) to narrow the window
    - performance dog!

Collecting Error Documentation

- Getting useful information
  - Use LE run-time option TERMTHDACT()
    - MSG Message only (default)
    - TRACE Traceback and message only
    - DUMP Basic CEEDUMP
    - UADUMP Full CEEDUMP, system dump
  - Use the LE run-time option TRAP(OFF)
    - not recommended in most situations

Collecting Error Documentation

- Messages
  - go to the console or can be directed to a file with MSGFILE Runtime option
- CEEDUMP
  - sent to the PRT
  - formatted dump of traceback and some storage
- System Dump
  - sent to the RDR
  - VMDUMP of storage - DUMPSCAN viewable

Collecting Error Documentation

- Getting **NON-USEFUL** information
  - When ABTERMENC(ABEND) is set the "original " ABEND is reissued at LE termination
    - **DO NOT TRAP ON THIS ABEND**
    - LE environment has already been cleaned-up and therefore a dump at this point is useless
    - Use TERMTHDACT to get the appropriate dump

Diagnosing Application Problems

'C' application example

Diagnostic Documentation

- Messages
- CEEDUMP
- System Dump
- Runtime Options Report
- Runtime Storage Report

```
#pragma runopts(trap(on) abtermenc(abend))

int first_func_called();          /* Function prototypes                */
int second_func_called();
int func_with_error();
int main(int argc, char *argv  ) /* Main Program                        */
 {
  int i = 0;
  i = first_func_called();        /* Call to "first_func_called()"      */
  return i;
 }
 int first_func_called()          /* Function: first_func_called()      */
 {
   int j;
   j = second_func_called();      /* Call to "second_func_called()"     */
   return j;
 }
 int second_func_called()          /* Function: second_func_called()    */
 {
   int k;
   k = func_with_error();          /* Call to "func_with_error()"       */
   return k;
  }
  int func_with_error()            /* Function: func_with_error()       */
  {
    int n = 0;
    int m;
    m = 10 / n;                    /* Force divide by 0 (ABEND0C9)      */
    return m;
  }
```

A Simple C program

```
#pragma runopts(trap(on) abtermenc(abend))

int first_func_called();        /* Function prototypes              */
int second_func_called();
int func_with_error();
int main(int argc, char *argv  ) /* Main Program                     */
 {
  int i = 0;
  i = first_func_called();      /* Call to "first_func_called()"    */
  return i;
 }
 int first_func_called()        /* Function: first_func_called()    */
 {
   int j;
   j = second_func_called();    /* Call to "second_func_called()"   */
   return j;
 }
 int second_func_called()       /* Function: second_func_called()   */
 {
   int k;
   k = func_with_error();       /* Call to "func_with_error()"      */
   return k;
  }
  int func_with_error()         /* Function: func_with_error()      */
  {
    int n = 0;
    int m;
    m = 10 / n;                 /* Force divide by 0 (ABEND0C9)     */
    return m;
  }
```

```
int n = 0;
int m;
m = 10 / n;
return m;
```

A Simple C program

Compile, Load, Execute

Ready; T=0.01/0.01 16:34:42
tcc markp *uses cc with the TEST(SYM) option*
Ready; T=0.03/0.06 16:35:05
load markp
Ready; T=0.01/0.01 16:35:18
genmod markp
Ready; T=0.01/0.01 16:35:23
markp
CEE3209S The system detected a fixed-point divide exception.
From compile unit MARKP at entry point func_with_error at statement 28 at compile unit offset +00000056 at address 01400426.

PRT FILE 0177 SENT FROM PICARD PRT WAS 0177 RECS 0587 CPY 001 I NOHOLD NOKEEP
DMSABE148T System abend 0C9 called from 01DFD386 reason code 00000009
CMS

CEEDUMP

CEE3DMP V1 R8.0: Condition processing resulted in the unhandled condition.

Information for enclave main

Information for thread 8000000000000000

Traceback:

DSA Addr	Program Unit	PU Addr	PU Offset	Entry	Service	Status
007FD018	CEEHDSP	01E18D28	+00002E94	CEEHDSP	UQ28585	Call
007FF378	MARKP	014003D0	+00000056	func_with_error		Exception
007FF2F0	MARKP	014002C8	+00000056	second_func_called		Call
007FF268	MARKP	014001C0	+00000056	first_func_called		Call
007FF1E0	MARKP	01400080	+00000056	main		Call
007FF0C8		01DDB22E	-01042D78	EDCZMINV		Call
007FF018	CEEBBEXT	00D9A3D8	+0000013C	CEEBBEXT		Call

CEEDUMP (cont)

```
Condition Information for MARKP (DSA address 007FF378)
   CIB Address: 007FD478
   Current Condition:
      CEE3209S The system detected a fixed-point divide exception.
   Location:
      Program Unit: MARKP Entry: func_with_error
      Statement:    28 Offset: +00000056
   Machine State:
       ILC..... 0004    Interruption Code..... 0009
       PSW.....  03EC2400 8140042A
       GPR0.....  007FF400     GPR1..... 01F67040   GPR2..... 81DDB2E2  GPR3.....  8140041A
       GPR4.....  80D9A4BC  GPR5..... 01400D8C  GPR6..... 00D969C8  GPR7.....   014010E8
       GPR8.....  00000001    GPR9..... 80000000   GPR10.... 81DDB222  GPR11.... 80D9A3D8
       GPR12.... 00D97A28   GPR13....007FF378   GPR14.... 00000000    GPR15....  0000000A
Storage dump near condition, beginning at location: 01400416
      +000000 01400416  18DE0530 4400C1B0 41E0000A 8EE00020  5DE03026 4400C1B8 58D0D00C
```

CEEDUMP (cont)

Condition Information for MARKP (DSA address 007FF378)

CIB Address: 007FD478

Current Condition:

CEE3209S The system detected a fixed-point divide exception.

Location:

Program Unit: MARKP Entry: func_with_error

Statement: 28 Offset: +00000056

Machine State:

ILC..... 0004 Interruption Code..... 0009

PSW..... 03EC2400 8140042A

GPR0..... 007FF400 GPR1..... 01F67040 GPR2..... 81DDB2E2 GPR3..... 8140041A

GPR4..... 80D9A4BC GPR5..... 01400D8C GPR6..... 00D969C8 GPR7..... 014010E8

GPR8..... 00000001 GPR9..... 80000000 GPR10.... 81DDB222 GPR11.... 80D9A3D8

GPR12.... 00D97A28 GPR13....007FF378 GPR14.... 00000000 GPR15.... 0000000A

Storage dump near condition, beginning at location: 01400416

+000000 01400416 18DE0530 4400C1B0 41E0000A 8EE00020 5DE03026 4400C1B8 58D0D00C

D r14,=F'0'

CEEDUMP (cont)

```
Parameters, Registers, and Variables for Active Routines:
CEEHDSP (DSA address 007FD018):
  Saved Registers:
    GPR0..... 00000000    GPR1.....   007FD3B4  GPR2.....  007FD478   GPR3..... 00
    GPR4..... 01F38330    GPR5.....   00000008   GPR6.....  81E1BBC0  GPR7..... 00
    GPR8..... 01E1BD25    GPR9.....   01E1AD26  GPR10.... 01E19D27  GPR11.... 01
    GPR12.... 00D97A28  GPR13.... 007FD018   GPR14.... 80D8F0E2   GPR15.... 81
GPREG STORAGE:
  Storage around GPR0 (00000000)
    +0000 00000000    Inaccessible storage.                                    . . . . . . . .133 wide
    +0020 00000020    Inaccessible storage.
    +0040 00000040    Inaccessible storage.
  Storage around GPR1 (007FD3B4)
    -0020 007FD394  007FD478 007FE017 01E1BD25 01E1AD26  01E19D27 01E18D28 0
    +0000 007FD3B4  007FE097 007FE0E7 007FDFA4 007FDFA4  007FDE04 007FDED4 0
    +0020 007FD3D4  007FD478 00000000 00000000 00000000  00000001 00000003 0

                                                  .
                                                  .
                                                  .
Additional Language Specific Information:

  errno information :
  Thread Id .... 8000000000000000 Errno ..... 0 Errnojr .... 00000000
```

System Dump

```
markp termthdact(UADUMP) abtermenc(ABEND) /
CEE3209S The system detected a fixed-point divide exception.
        From compile unit MARKP at entry point func_with_error at statement 28
at compile unit offset +00000056 at
        address 01400426.
PRT FILE 0181 SENT FROM PICARD   PRT WAS 0181 RECS 0987 CPY  001 1 NOHOLD NOKEEP
DMSABE148T System abend 0C9 called from 01DFD386 reason code 00000009
CMS
```

Notes:

- System dump is then produced
- DUMPLOAD to load the dump
- DUMPSCAN to analyze

System Dump (cont)

```
 REGS
CPU ADDRESS - 0000                    PREFIX REGISTER - 00000000
GENERAL REGS  0 - 15
    00000000  00E0F3B4   01DFB0A4 00E11488   40000030  00000018 00E0F478   00E10017
    01DF3D25 01DF2D26 40000000  81DFAF70  00E3CA28 00E11400 81DF33BC 81DFAF70
CONTROL REGS  0 - 15
    000010E0 00000000 00000000 00000000  00000000 00000000 FF000000 00000000
    00000000 00000000 00000000 00000000  00000000 00000000 DF000000 00000000
ACCESS  REGS  0 - 15
    00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
FLOATING POINT REGS  0 - 6
    4E000000 002BC83D 00000000 00000000  00000000 00000000 00000000 00000000

TOD CLOCK                B415D20A 03C03D48            PSW 03EC0400 81DFB010
CLOCK COMPARATOR  00000000 00000000
CPU TIMER                FFFFC774 10B470B1

EXT OLD   030C0000 00F34C10          INT CODE 4001              EXT NEW 00080000 80F22D08
SVC OLD   03EC1400 80E45F52          INT CODE 00CC ILC 0002     SVC NEW 000C0000 80F111DE
PGM OLD  03EC2400 81DFD386           INT CODE 0009 ILC 0004     PGM NEW 00080000 80F3E9C0
MCH OLD 000C2000 80F120D4            MCH NEW 00080000 80F3E310
```

System Dump (cont)

- PSW and Regs from System Dump are useless for debug
- use data from CEEDUMP for display
  - D 140042A -from CEEDUMP PSW

```
01400420  000A8EE0  00205DE0  30264400  C1B858D0 E6   *......).....A..........*
01400430  D00458E0  D00C9834 D020051E  07070000       *......q................*
01400440  00000000  01400458  00000000  014002C8      *..... ....... .H.......*
01400450  00000001  00000000  014003D0  0000006C      *........ .....%........*
01400460  0140048E  00000000  0320004A  00000320      *. .......................*
01400470  004F0000  0380805B  000003A0  005A0000      *.|.....$...............*
01400480  03C0005A  000003C0 005E0000  03C00E0E       *........;................*
01400490  0E0E0E0E  10CEA186 FFFFFD24  00000074       *......f..................*
014004A0  00000000  FE000000  00000000  90000000      *........................*
014004B0  02C00019  000F86A4  95836DA6  89A3886D      *.....func_with_*
014004C0  85999996  99405000  0036FFFF  FF3C3825      *error &...........*
```

- Module names come **at the end** of the code

RunTime Options Report

- markp termthdact(quiet) rptopts(on) msgfile (rpts) /

```
1Options Report for Enclave main 05/23/00 2:40:10 PM

LAST WHERE SET          OPTION
-----------------------------------------------------------------------
Installation default      ABPERC(NONE)
Programmer default        ABTERMENC(ABEND)
Installation default    NOAIXBLD
Installation default      ALL31(OFF)
Installation default      ANYHEAP(16384,8192,ANYWHERE,FREE)
Installation default    NOAUTOTASK
Installation default      BELOWHEAP(8192,4096,FREE)
Installation default      CBLOPTS(ON)
Installation default      CBLPSHPOP(ON)
Installation default      CBLQDA(ON)
Installation default      CHECK(ON)
Installation default      COUNTRY(US)
Installation default      DEBUG
Installation default      DEPTHCONDLMT(10)
Installation default      ENVAR("")
Installation default      ERRCOUNT(0)
Installation default      ERRUNIT(6)
Installation default      FILEHIST
```

RunTime Options Report (cont)

Default setting	NOFLOW
Installation default	HEAP(32768,32768,ANYWHERE,KEEP,8192,4096)
Installation default	HEAPCHK(OFF,1,0)
Installation default	HEAPPOOLS(OFF,8,10,32,10,128,10,256,10,1024,10,2048,1
Installation default	INQPCOPN
Installation default	INTERRUPT(OFF)
Installation default	LIBRARY(SYSCEE)
Installation default	LIBSTACK(8192,4096,FREE)
Invocation command	**MSGFILE(rpts,FBA,121,0,NOENQ)**
Installation default	MSGQ(15)
Installation default	NATLANG(ENU)
Installation default	NONONIPTSTACK(4096,4096,BELOW,KEEP)
Installation default	OCSTATUS
Installation default	NOPC
Installation default	PLITASKCOUNT(20)
Installation default	POSIX(OFF)
Installation default	PROFILE(OFF,"")
Installation default	PRTUNIT(6)
Installation default	PUNUNIT(7)
Installation default	RDRUNIT(5)
Installation default	RECPAD(OFF)

RunTime Options Report (cont)


```
 Invocation command        RPTOPTS(ON)
Installation default      RPTSTG(OFF)
Installation default    NORTEREUS
Installation default      RTLS(OFF)
Installation default    NOSIMVRD
Installation default      STACK(131072,131072,BELOW,KEEP)
Installation default      STORAGE(NONE,NONE,NONE,8192)
Invocation command        TERMTHDACT(QUIET)
Installation default    NOTEST(ALL,"*","PROMPT","INSPPREF")
Installation default      THREADHEAP(4096,4096,ANYWHERE,KEEP)
Installation default      TRACE(OFF,4096,DUMP,LE=0)
Programmer default        TRAP(ON,SPIE)
Installation default      UPSI(00000000)
Installation default    NOUSRHDLR()
Installation default      VCTRSAVE(OFF)
Installation default      VERSION()
Installation default      XUFLOW(AUTO)
```

RunTime Storage Report

- markp termthdact(quiet) rptstg(on) msgfile (rpts) /

1Storage Report for Enclave main 05/23/00 2:56:48 PM

STACK statistics:	
Initial size:	131072
Increment size:	131072
Maximum used by all concurrent threads:	4048
Largest used by any thread:	4048
Number of segments allocated:	**1**
Number of segments freed:	0
NONIPTSTACK statistics:	
Initial size:	0
Increment size:	0
Maximum used by all concurrent threads:	0
Largest used by any thread:	0
Number of segments allocated:	0
Number of segments freed:	0
LIBSTACK statistics:	
Initial size:	8192
Increment size:	4096
Maximum used by all concurrent threads:	8000
Largest used by any thread:	8000
Number of segments allocated:	1

RunTime Storage Report (cont)

- markp termthdact(quiet) rptstg(on) msgfile (rpts) /

HEAP statistics:

Initial size:	32768
Increment size:	32768
Total heap storage used (sugg. initial size):	**3128**
Successful Get Heap requests:	**10**
Successful Free Heap requests:	**4**
Number of segments allocated:	**1**
Number of segments freed:	0

Summary

- CEE32xx messages are really ABEND0Cy.
- Don't Trap on an LE reissued ABEND!
- Change runtime options to get a dump
- LE reported errors need further investigation to determine owner of **original** problem.
  - GET PSW and Regs from CEEDUMP
- Use **EXCEPTION** line in CEEDUMP for program checks.

Additional Information

- Program Directory
- LE Programming Reference
- LE Programming Guide
- LE Migration Guide
- All LE documentation available on
  - VM collection CDs
  - LE website
    - http://www.ibm.com/s390/le